

A Structured Approach for Reviewing Architecture Documentation

Robert L. Nord, Software Engineering Institute
Paul C. Clements, Software Engineering Institute
David Emery, DSCI
Rich Hilliard, consulting software systems architect

December 2009

TECHNICAL NOTE
CMU/SEI-2009-TN-030

Research, Technology, and System Solution Program
Unlimited distribution subject to the copyright.

<http://www.sei.cmu.edu>



This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2009 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. This document may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

Table of Contents

| | |
|---|------------|
| Acknowledgments | vii |
| Abstract | ix |
| 1 Introduction | 1 |
| 2 Conceptual Basis for the Approach | 3 |
| 2.1 WHY: Purpose | 3 |
| 2.1.1 Review for AD conformance to a normative specification | 3 |
| 2.1.2 Review for AD suitability to support use of the architecture for its intended purpose | 3 |
| 2.1.3 Review for AD suitability to support architecture evaluation or analysis | 4 |
| 2.2 WHO: Stakeholders | 5 |
| 2.3 WHAT: What Is Reviewed | 6 |
| 2.4 WHEN: AD Reviews in the Life Cycle | 7 |
| 3 Steps of the Approach | 8 |
| 4 Question Sets for Reviewing the AD | 11 |
| 4.1 Sample Question Set for Capturing the Right Stakeholders and Concerns | 12 |
| 4.2 Sample Question Set for Reviewing Choice of Framework and Viewpoints | 14 |
| 4.3 Sample Question Set for Supporting Evaluation | 16 |
| 4.4 Sample Question Set for Supporting Development | 18 |
| 4.5 Sample Question Set for Identifying Architecturally Significant Requirements and Key Design Decisions | 22 |
| 4.6 Sample Question Set for Reviewing for Conformance to ISO/IEC 42010 | 25 |
| 5 Examples of Constructing a Review | 28 |
| 5.1 An Example: AD Reviews in the Context of the ATAM | 28 |
| 5.2 An Example: AD Reviews in the Context of SARA | 30 |
| 5.3 Building Reviews from Question Sets | 32 |
| 6 Related Work | 34 |
| 7 Results and Next Steps | 37 |
| Appendix ISO/IEC 42010 | 39 |
| References/Bibliography | 43 |

List of Figures

| | | |
|-----------|--|----|
| Figure 1: | Notional Life-Cycle Stages | 7 |
| Figure 2: | Template for a Question Set | 12 |
| Figure 3: | ATAM Phase 0 Go/No-Go Criteria | 28 |
| Figure 4: | Core Concepts of ISO/IEC 42010:2007 | 39 |
| Figure 5: | Additions to the Core Concepts of ISO/IEC 42010:2007 | 41 |

List of Tables

| | | |
|----------|---|----|
| Table 1: | Common Stakeholder Roles | 5 |
| Table 2: | Building an ATAM AD Review from Question Sets | 29 |
| Table 3: | Reviewing Architecture Documentation as a SARA Method | 32 |
| Table 4: | Architecture-Related Reviews and Their Associated Question Sets | 33 |

Acknowledgments

The authors wish to thank the members of the *Workshop on Reviewing Architecture Descriptions*, held at the 2008 Working IEEE/IFIP Conference on Software Architecture (WICSA 2008), for providing feedback on a preliminary draft of this document. Paris Avgeriou (University of Groningen) and Patricia Lago (Vrije University) used the material in graduate courses at their universities and their report on the experiences of their students helped us better understand the application of the approach. In their careful reviews, Larry Jones, Len Bass, James Ivers, and Linda Northrop provided comments that greatly improved this technical note.

Abstract

This technical note proposes a structured approach for reviewing architecture documentation. Given the critical importance of architecture to software project success, it follows that the architecture cannot be effective unless it is effectively captured in documentation that allows the architecture's stakeholders to understand and use the architecture in the way it was intended. The approach does not assume a particular architecture methodology or a particular architecture documentation practice, although it was conceived in the context of the International Organization for Standardization (ISO) Recommended Practice for Architecture Description of Software-Intensive Systems and the SEI Views and Beyond approach to documenting software architectures. Like both of them, our approach is centered on the stakeholders of the artifact, engaging them in a focused, guided way to ensure that the documentation carries sufficient quality to enable them to do their jobs and to help them point out gaps and weaknesses. Our approach is not intended as a complete framework for architecture evaluation; rather it is meant to be used within such a framework, when one is available.

1 Introduction

This technical note proposes a structured approach for reviewing architecture documentation (AD).¹ Given the critical importance of architecture to software project success, it follows that the architecture cannot be effective unless it is effectively captured in documentation that allows the architecture's stakeholders to understand and use the architecture in the way it was intended. That is, the *documentation* of the architecture inherits the criticality of the architecture itself. An architecture that cannot be understood (or, worse, is misunderstood) because of deficient documentation will fail to meet its goals as surely as a poorly chosen architecture.

Put succinctly, *if your architecture is not well described, it doesn't matter if it's well designed.*

To be clear, we are not discussing an approach for evaluating an *architecture*; there are several existing methods for that already [SARA 2002, Bass 2003, Clements 2002, Dobrica 2002, Babar 2004]. Rather, we are proposing an approach for evaluating the *documentation* of an architecture (one purpose of which may be to support an architecture evaluation exercise).

We call our proposal an approach, not a method, because a method would come with a fully worked-out process model and a rich pedigree of usage that would inform detailed guidance at each step. We have neither of those, even though the conceptual heart of the approach is rooted in well-grounded practical methods with a long history of successful use. Rather, we feel that the approach is a starting point for a method. Moreover, there are a number of existing methods (see Section 6 “Related Work” on page 34) that could serve as frameworks for fleshing out the details of the present approach for use within those methods. The approach could be used as a stand-alone review, but it is more likely to be used in conjunction with or in the support of other software development life-cycle activities. As with other forms of analysis, the approach could be used proactively to provide guidance on what to document as a portion of each design/documentation step as well as be used as a separate review activity after much of the documentation has been completed. The focus of this report is on the latter aspect.

The approach does not assume a particular architecture development methodology or a particular architecture documentation approach, although it was conceived in the context of ISO/IEC 42010 (ISO adoption of ANSI/IEEE 1471:2000) [ISO/IEC 42010:2007] (looking forward to concepts proposed for revision as described in Appendix) and the SEI Views and Beyond approach to documenting software architectures [Clements 2003]. Like both of them, our approach is centered on the stakeholders of the artifact, engaging them in a focused, guided way to assure that the documentation carries sufficient quality to enable them to do their jobs and to help them point out gaps and weaknesses.

Just as it does not assume a particular methodology, neither does the approach assume a particular form of the artifact being reviewed. By *documentation*, we mean hard- or soft-copy written and graphic materials that describe or specify the architecture of a software-intensive system.

¹ The preferred term in the SEI Views and Beyond Approach [Clements 2003] is *documentation*, whereas the preferred term in ISO/IEC 42010 [ISO/IEC 42010 2007] is *description*. Each side has stated reasons for its choice, but the distinction is unsubstantial and beyond the scope of this report. We chose the term *documentation* as the object of the review approach presented herein to be consistent with the terminology used in other SEI publications.

The need for AD reviews arises across a spectrum of situations throughout the life cycle of a system, all involving using the documentation for some specific purpose. These include (among many others)

- using the architecture as the basis for downstream design or implementation of components or subsystems identified by the architecture
- checking to see if design or implementation conforms to the architecture
- seeing if the architecture is ready to support an evaluation milestone for fitness of purpose
- using the documentation to support project planning (such as assigning modules to teams or planning incremental deliveries)

This technical note is organized as follows:

- Section 2 establishes the conceptual basis for the approach, which we give in terms of *why* (the purpose of the review), *who* (who is involved in the review), *what* (the artifact or artifacts reviewed) and *when* (where in the life cycle the review may occur).
- Section 3 refines this four-part conceptual basis into a six-step approach.
- Section 4 introduces a number of specific question sets that can serve as examples to satisfy some of the purposes laid out earlier.
- Section 5 discusses using the approach in practice, with a walked-through example.
- Section 6 gives related work.
- Section 7 discusses results and next steps.

2 Conceptual Basis for the Approach

The approach proposed in this note embodies a number of assumptions. For clarity, we will enumerate and discuss the important ones. This discussion will, in fact, lead quickly to the step-by-step details of our approach. The conceptual basis can be summarized conveniently as dealing with the *why*, *who*, *what*, and *when* of AD reviews.

2.1 WHY: Purpose

Reviews are conducted for some purpose—that is, to see if the AD can be used for some purpose.

One of the truisms of architecture evaluation is that architectures are neither inherently good nor bad, but only well-suited or not with respect to a particular set of goals. The same applies to AD reviews. The AD is neither inherently good nor bad, but only sufficient or not with respect to an anticipated use. An AD review, therefore, requires identifying the purpose of the review.

Below are three examples of why the AD might be reviewed.

2.1.1 Review for AD conformance to a normative specification

This kind of review is intended to discover if the AD conforms to some normative specification that has been imposed on it. The focus is on the AD itself; the architecture it describes is de-emphasized. For example

- If the AD claims conformance to ISO/IEC 42010:2007, does it conform to the requirements (“shalls”) of that standard?
- If the project is using an architecture framework such as the Department of Defense Architecture Framework (DODAF), The Open Group Architecture Framework (TOGAF), or the Federal Enterprise Architecture Framework (FEAF), does the AD conform to its framework?
- Does the AD conform to (is it consistent with) other standards, guidelines, or templates mandated by the developing organization or by the client?

2.1.2 Review for AD suitability to support use of the architecture for its intended purpose

This kind of review is carried out to see if stakeholders of the architecture can use the AD to do their jobs. The focus is on how well the AD describes the architecture. Understandability and usability of the AD are important review criteria. Examples include

- Can the AD support downstream software design and development? Can the AD enable effective communications among organizations involved in the development, production, fielding, operation, and maintenance of a system? Here, important concerns are comprehension and completeness, as well as the precise conveyance of global design concepts so that all groups have the same mental model of the architecture.
- Can the AD support certifying conformance of downstream designs and implementations to the architecture?

- Can the AD serve as input to system generation and analysis tools? Concerns here include completeness and perhaps the machine readability of particular sections and conformance of those sections to a specification language.
- Can the AD support system evolution in concert with the architecture and the associated business planning for evolution? Here, the focus is on design rationale and the explanation of architectural drivers.
- Can the AD support project planning, budgeting, and scheduling? Here, the emphasis is on the ability to predict the size, complexity, risk, reuse opportunities, and requirements for specific expertise.
- Can the AD support the organizational structure of the development team? Here, the emphasis is on allocation views that document work assignments and the implementation environment.
- Can the AD support the development of a group of systems sharing a common set of features and built from a common set of core assets? Here, the emphasis may be on the specification in the AD of commonalities, points of variation, and variation mechanisms built into the architecture.
- Can the AD support the creation of maintenance and training materials?
- Can the AD support communications between acquirers and developers as a part of contract negotiations? Can the AD support preparation of acquisition documents (e.g., requests for proposal and statements of work)?
- Can the AD support analysis of interfaces of other systems and their architectures? Here, the emphasis may be on interoperability or the relationship with other dependent architectures within an enterprise.

2.1.3 Review for AD suitability to support architecture evaluation or analysis

This kind of review is carried out to see if the AD provides sufficient information to be able to predict system qualities by examining or analyzing the architecture. Examples include

- Can the AD support an architecture evaluation using a method such as the SEI Architecture Tradeoff Analysis Method[®] (ATAM[®]) [Clements 2002]? Here, important concerns are attention to architecturally significant requirements, the quality attributes required of and provided by the architecture, as well as evidence of feasibility—namely, that the architecture can, in fact, be built under the budget and schedule allotted.
- Can the AD support analysis of alternative architectures? The AD must have the qualities necessary to evaluate an architecture by itself but also include sufficient information about key design decisions and their rationale to provide in-depth qualitative insight about whether the architecture is well-suited to take the organization into the future, so it can be compared with other candidates.

[®] Architecture Tradeoff Analysis Method and ATAM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Prerequisite to an AD review, then, is knowing for what purpose the AD must be suitable. How do we know what purpose or purposes the AD must serve (and hence for what purpose or purposes it should be reviewed)? The stakeholders of the architecture and its AD will tell us.

2.2 WHO: Stakeholders

The AD is used by one or more stakeholders to achieve a specific purpose, such as completing a task assigned to a stakeholder.

There is no fixed, complete list of stakeholders, either for the AD or for the architecture of the system under review. For example, in safety-critical systems, the safety analyst is one of the most important stakeholders, as this person can often halt development on the spot if he or she believes the design has let a fault slip through. In most IT systems, there is no such role, although IT systems might have a stakeholder whose job is to make sure that the system does not violate any accounting rules or financial statutes.

Table 1 below lists a set of notional stakeholders of the architecture of the system who, in our experience, commonly have a keen interest in using the AD to carry out their jobs. The table is by no means complete but rather meant to be illustrative. It is derived from information about potential stakeholders in ISO/IEC 42010 [ISO/IEC 42010:2007]. An architecture (and hence, the AD) in your organization may well have stakeholders not in this list or have a different name for a stakeholder in this list. (One person, group, or organization may be responsible for more than one stakeholder role.)

Table 1: Common Stakeholder Roles

| Name | Description |
|----------------------|---|
| Architecture Analyst | Responsible for analyzing the architecture to make sure it meets certain critical quality attribute requirements. Analysts are often specialized; for instance, performance analysts, safety analysts, and security analysts are often well-defined positions in a project. |
| Architect | Responsible for the development of the architecture and its documentation. Focus and responsibility is on the system under review. |
| Business Manager | Responsible for the functioning of the business/organizational entity that owns the system under review. Includes managerial/executive responsibility, responsibility for defining business processes. Able to assess the ability of the architecture to meet business goals. |
| Customer | The client who pays for the system and insures its delivery. The customer often speaks for or represents the end user, especially in a government acquisition context. |
| Designer | Responsible for systems and/or software design, applying the architecture to meet specific requirements. |
| Evaluator | Responsible for conducting an evaluation of the architecture (and its documentation) against some clearly defined criteria. |
| Fielder | Responsible for accepting the completed system from the development effort and deploying it, making it operational and fulfilling its allocated business function. |
| Implementer | Responsible for the development of specific components (e.g., software modules) according to designs, requirements, and the architecture. |
| Integrator | Responsible for taking individual components and integrating them, according to the architecture and system designs. |
| Maintainer | Responsible for fixing bugs and providing enhancements to the system through its use (including adaptation of the system for uses not originally envisioned.) |
| Software Manager | Responsible for planning, sequencing, scheduling, and allocating resources (particularly architects, designers, implementers, integrators, and testers) to develop software components and deliver components to integrators and testers. |
| System/Program | Responsible for planning, sequencing, scheduling, and allocating resources (particularly architects and designers) to develop system components (less software components) and |

| Name | Description |
|---------|--|
| Manager | integrate them into a system provided to fielders for operational use, deployment, and acceptance. |
| Tester | Responsible for the (independent) test and verification of the system or its components against the requirements and the architecture. |
| User | The actual end users of the system. There may be distinguished kinds of users, such as administrators, network administrators, super-users, and so on. |

Prerequisite to an AD review, then, is the identification of the AD stakeholders. The AD stakeholders include the architecture's stakeholders and possibly others. If the purpose of the review is AD conformance, the stakeholders should include people who are familiar enough with the normative specification to render a judgment.

If the purpose of the review is to see if the AD makes the architecture usable for its intended purpose, the stakeholders involved in that use should be represented in the AD review.

Finally, if the purpose of the review is to see if the AD supports architecture evaluation or analysis, the stakeholders should include the evaluation or analysis team.

2.3 WHAT: What Is Reviewed

A number of artifacts need to be available and present for an AD review to take place. Obviously, the AD must be provided. The purpose of the review will determine the additional artifacts the reviewers need to understand the criteria they will use to review the AD. For example, if the AD is being reviewed for conformance to a standard or to a framework, the normative requirements of the standards/framework must also be available.

The AD may or may not exist as a single document or artifact. Some projects produce a series of "architecture notes" documenting successive decisions. The architecture is captured by this series. Other projects manage an evolving AD in a repository. In all cases, exactly what constitutes the AD should be determined before the review begins, so that all reviewers work from exactly the same description of the architecture.

Many of the AD review purposes do not require a complete AD to be present. Therefore, the "what" aspect of AD reviews can be planned in terms of sections and subsections, as opposed to the entire document. Reviewing parts of the document as soon as they become available can provide for scheduling flexibility and identify some problem areas much earlier. It is often very fruitful to review part of the AD early (before the whole AD is released) to identify problems and risks early and to ensure that the AD is well-positioned to meet its stakeholders' expectations.

A framework review (see Section 4.2 on page 14) can be undertaken before the AD is begun. A framework review assesses not the architectural content of the AD but rather the set of viewpoints, model types, and tools that are to be used in preparing that AD to determine whether the architect has selected the right tools for the job.

2.4 WHEN: AD Reviews in the Life Cycle

Finally, a project milestone for each AD review could be defined for each AD review to record the progress of architecture development. This is most appropriately done in the context of a specific life-cycle model being used by the project organization, such as those found in ISO 12207 [ISO/IEC 12207:2008] (for software engineering), ISO 15288 [ISO/IEC 15288:2008] (for systems engineering), the Rational Unified Process [IBM 2004], or agile project management approaches such as Scrum [Schwaber 2004]. For the sake of concreteness in this technical note, the simplified life cycle described in ISO TR 24748 [ISO/IEC CD TR 24748:2007] is used. See Figure 1.

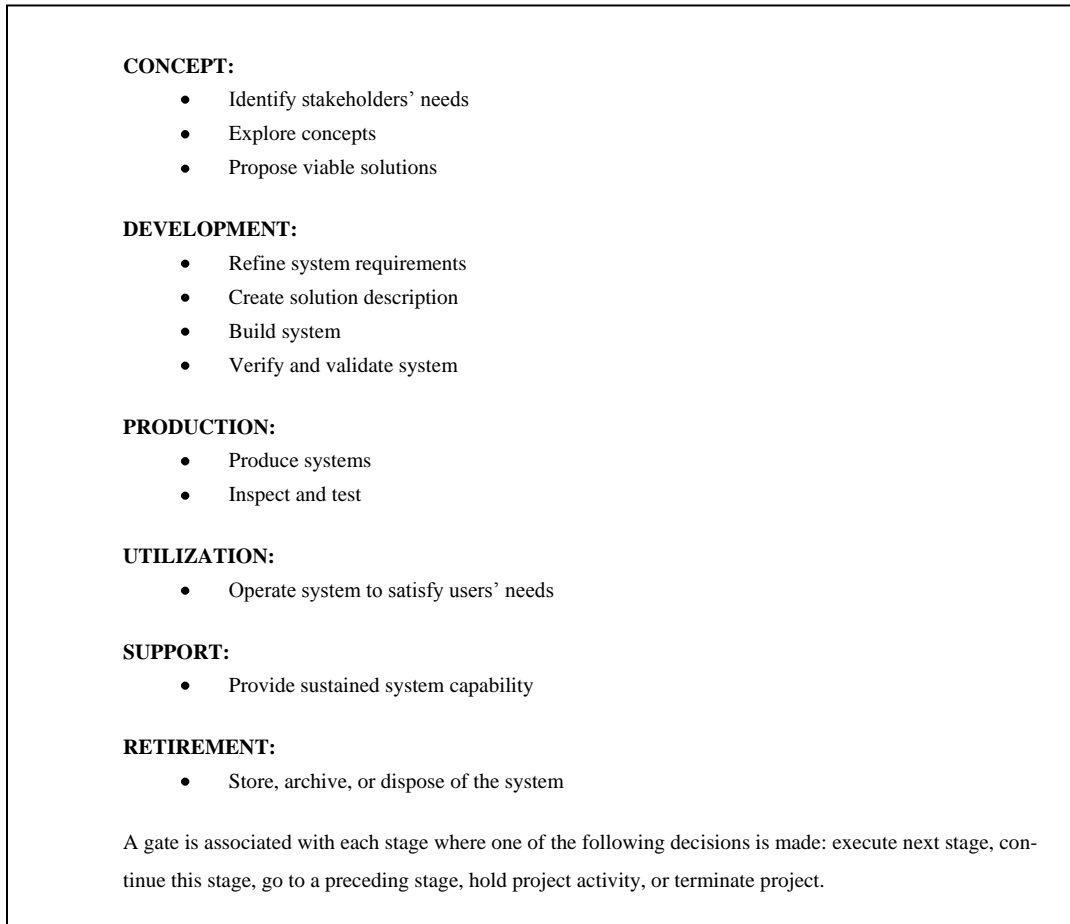


Figure 1: Notional Life-Cycle Stages

Various classes of life cycle models comprising stages have been described including waterfall, iterative and incremental development, evolutionary development, and spiral.

Various AD reviews are most appropriate at different life-cycle stages. See Table 4 on page 33 for examples.

3 Steps of the Approach

From the conceptual basis given in Section 2, we can posit a set of steps that constitute a methodical AD review. The steps are led by the AD review team with support from the AD stakeholders. Those steps are

- **Step 1: Establish the purpose of the review.** An AD review establishes whether the AD is fit for some specific purpose by a set of identified stakeholders. The purpose of an AD review may come from the life-cycle development model in use; the AD may be reviewed specifically to see if it can support an upcoming development activity. Stating that purpose will focus the review participants and establish a basis for whether the review activity is achieving the desired goal. Section 2.1 lists a number of possible (and usual) purposes for reviewing the AD; choose one or more of them or craft your own. A review purpose can be stated as a scenario that describes how a particular stakeholder can successfully use the AD to carry out part of his or her job. The goal and expected outcomes for the review should also be described as part of this step. The expected outcome of the review could include a go/no-go decision about whether the desired state of the AD has been achieved and a list of problems in the AD that prevent the stakeholders from using it in successfully performing their jobs.

It is likely that any AD will need to be fit for more than one purpose, and hence the review will be multi-faceted (the alternative is several smaller reviews, each with a single purpose). Thus, you should expect to write down a collection of purposes, possibly as scenarios. The scenarios would describe a stakeholder using the AD for some purpose.

Identifying the review purpose will also identify the stakeholders who should be represented in the review, so the stakeholders for the review should be explicitly listed as part of this step. Section 2.2 lists a number of notional stakeholders who have an interest in using the AD.
- **Step 2: Establish the subject of the review.** An AD review requires a number of artifacts to be available. This step involves identifying the types of artifacts, the version of the artifacts, their sources, and the degree of completeness of the artifacts necessary to conduct the review. Section 2.3 lists a number of artifacts that can be reviewed—the AD, an architecture framework (in the sense of TOGAF or DODAF), one or more viewpoint definitions, and so on. Choose one or more of them or add your own. Use the purpose(s) laid out in Step 1 to establish the artifact collection required and then gather them for the review.
- **Step 3: Build or adapt the appropriate question set(s).** This step involves identifying the questions that your review will put to the AD. If you already have a set of questions that meets the purpose of your review, you can use it (perhaps with some modification). If not, you will have to construct it. Organizing questions as question sets allows them to be reused by providing contextual information about the purpose and stakeholder concerns that need to be addressed as well as guidance for obtaining and interpreting the results. Section 4 discusses question sets in more detail. If you chose to use existing questions sets, they must be tailored for the purposes of the review. Questions that are not relevant can be omitted; for example, some questions may not be appropriate when reviewing the AD early in the life

cycle (before much architecting has been done). General questions can be made more specific according to the technology of the project (e.g., references to data persistence may be replaced by references to Oracle database). The question set(s) that you pick will suggest a particular approach, and the questions need to be formulated appropriately. For example, will you use the Active Design Review technique [Parnas 1985], a questionnaire or checklist given to stakeholders, some sort of automated or measurement-based analysis, or some other approach? (Active design reviews are explained in Section 6, “Related Work” on page 34.)

- **Step 4: Plan the details of the review.** Different AD reviews are appropriate at different life-cycle stages. The purpose of the review will constrain when in the life cycle it would be most advantageous to carry out the review (as discussed in Section 2.4). This step involves the timeframe and the basic format of the review. The timeframe might allow as much time as needed to answer questions or only a limited amount of time (e.g., answers are time boxed). Limited time requires prioritizing the questions according to the goals of the review. Time and resources will affect the format and “weight” of the review (e.g., just in time, triage, or full reviews). How the results will be communicated needs to be determined and could affect the format and weight of the required answers.

This step also involves identifying the actual review participants (not just abstract stakeholder roles) and securing their participation. An initial assignment of questions to the reviewers responsible for asking them and the stakeholders responsible for supplying the answers can be made at this time. As the review is conducted, the initial priorities and stakeholder assignments may change as a deeper understanding of the documentation is gained and the reviewers probe further into applicable areas.

This step also involves handling the logistics for the review—time and place of meeting(s), paying for everyone’s time, providing read-ahead materials, and so on.

- **Step 5: Perform review.** Performing the review involves posing the questions to the stakeholders involved in the review and gathering their answers. Depending on the specific approach chosen, this might involve an individual objective review where stakeholders also play the role of the reviewer and pose questions to themselves or an inspection where a separate review team poses questions to the stakeholders. Inspections could take the form of an all-hands gathering, a number of one-on-one meetings, or something in between; these meetings could be face-to-face or distributed and remote (e.g., email, online virtual meetings, or a content management system such as a wiki). After the results are gathered, the evaluation considerations and criteria are applied, as defined by the chosen question set(s). Although the reviewers can make some preparations, not all the important issues can be known a priori. These issues must be determined in the initial part of the review and will influence the questions and artifacts used as the reviewers dig deeper in these areas.
- **Step 6: Analyze and summarize results.** The intent is to aggregate the answers to the questions and then make a qualitative determination of the overall impact of the AD against the stakeholders and concerns. Results are not likely to be a simple pass/fail but rather a more nuanced conclusion concerning specific problems in specific parts of the AD.

The steps outlined above are described in terms of a “standalone review” focused specifically on the AD, but this approach can be used in other ways. The steps can support an activity to review architecture documentation as a standalone event, or the steps can be part of a larger review of the

architecture itself (e.g., using the ATAM or Software Architecture Review and Assessment framework [SARA 2002]). Review question sets can be used as gating criteria by technical management to establish the maturity of the AD. (It would be effective to tie such an approach to a program's Earned Value Management System [ANSI/EIA-748-B 2007] for architecture, where the completion of each kind of review is associated with a specific amount of earned value.) The review question sets can serve as checklists for the architects to use in the production and maturation of the AD. A checklist produced from AD review questions sets could be used by external actors such as an organizational Quality Assurance group to assess the maturity of the AD or by a customer as a set of acceptance criteria for the delivery of the AD.

4 Question Sets for Reviewing the AD

Posing and answering questions in a review is, of course, the heart of the matter. This section discusses what is involved in the formation of question sets—groups of questions that, together, address a narrowly focused purpose for an AD review.

First of all, question sets can and should be reused where possible and appropriate. Review questions that have been carefully crafted to address a specific purpose constitute an investment of time and effort, and this investment can be amortized over every review in which the question applies.

Like all artifacts that are designed to be reused, a question set requires some ancillary information to facilitate that reuse. Besides the questions themselves, a question set must also contain information that allows a user to make sure the question set is appropriate and to use it effectively, as shown below:

1. **Question set name:** As an artifact to be reused, it is very helpful to give the question set a name by which it can be referred. A concise statement of the purpose can often be useful to capture in the name; for example, “ISO/IEC 42010 conformance” or “Ready to support an evaluation using the ATAM.”
2. **Purpose:** What purpose does the question set address? Section 2.1 lists some specific review purposes. (An AD review may well have more than one purpose, which means that more than one question set may be involved. There is a balance to be struck between holding a number of small, separate reviews, each focused sharply on one narrow purpose, and the expediency of holding a single all-encompassing review.)
3. **Stakeholders and concerns:** Who are the stakeholders, and which of their concerns are being addressed by the question set? (See Section 2.2 on page 5.) Making stakeholders and concerns a first-class dimension of an AD review effectively elaborates the purpose of the question set and informs the formulation of the questions.
4. **Questions:** This section contains the questions that constitute the question set. For each question, the following information applies:
 - a. **Respondents:** To whom should each question be posed? The questions might be addressed to the person speaking for the AD. Usually this will be the architect. The questions might be addressed to reviewers checking the understandability of the AD by using it to answer questions about the architecture it describes. For instance, if the AD should support project planning (a purpose) and is being reviewed for such (using a “project planning” question set), the respondents would include those concerned with project planning—technical managers. If the AD should support development and is now being reviewed for that, the respondents will certainly include key developers. Questions about the AD itself can be answered by examining the AD or analyzing it with a tool (for example, automatically checking to make sure that every cross-reference is defined).

The person(s) to whom a question is posed may or may not be the same as the stakeholder(s) whose concern the question addresses. Review participants may be proxies for stakeholders.

- b. **Expected answers:** How do we evaluate the answer(s)? The questions should come with the expected answers and a set of considerations and criteria to help the reviewers evaluate the AD based on the answers they receive. For example, they might wish to understand not just the answers given by the reviewers but also how much difficulty the reviewers had coming up with those answers. They might wish to understand the criteria the stakeholders used for why they answered “yes, we’re happy” or “no, we’re not happy.”

The respondents can be shown the criteria so they can more fully answer the questions; however, they should not be shown the expected answers, to avoid biasing their answers.

- c. **Criticality:** How critical is each question? The “wrong” answer to some questions might halt a project until it’s resolved, whereas the “wrong” answer to other questions might merely be something to watch over time. The questions should come with guidance (perhaps a weighting) to help establish their importance.
5. **Advice:** The question set should provide additional useful information on how and when the review should be conducted. This section might relate experience gained through using the question set in a prior review.

Figure 2 provides a template that can be used when constructing a question set.

| | | | |
|--------------------------------------|------------------------|-----------------------------|------------------------|
| 1. Question set name: | | | |
| 2. Purpose: | | | |
| 3. Stakeholders and concerns: | | | |
| 4. Questions | 4a. Respondents | 4b. Expected answers | 4c. Criticality |
| 5. Advice: | | | |

Figure 2: Template for a Question Set

Following are a few sample question sets to serve specific AD review purposes. They are written in different styles to illustrate the ways a question set may be used. For example, the sample question set for capturing the right stakeholders in Section 4.1 is written in the active design review style, and the questions are really directions to stakeholders to use the AD for some purpose. The other sample question sets are written as if an interviewer is questioning a stakeholder. These could be adapted to an active design review style or for the purposes of an individual objective review. Some questions that can be answered yes or no are serving as filters, and when the answer is yes, it is appropriate to ask follow-up questions of the form, “How do you know?”

4.1 Sample Question Set for Capturing the Right Stakeholders and Concerns

Every AD in compliance with ISO/IEC 42010-2007 [ISO/IEC 42010:2007] is required to explicitly list the stakeholders and stakeholder concerns addressed by the architecture. The Views and Beyond approach to architecture documentation uses the explicit identification of stakeholders and their concerns to determine which views to include in the AD [Clements 2003]. Therefore, a

useful review of the AD examines its choice of stakeholders and concerns to insure that the important ones are accounted for. Such a review could be usefully carried out quite early, when the stakeholders and concerns are documented but before the rest of the AD is created.

The questions in the sample question set below are formulated using the Active Design Review technique [Parnas 1985]. Active design reviews are explained in Section 6, “Related Work” on page 34.

| 1. Question set name: Capturing the right stakeholders and concerns | | | |
|--|------------------|--|---|
| 2. Purpose: Use this question set to gauge the appropriateness of the architect’s list of stakeholders and concerns for completeness, over-completeness, and appropriateness, and to review how well the stakeholders believe their interests and concerns have been captured. | | | |
| 3. Stakeholders and concerns: All those with a substantial stake in the architecture should be involved or have their roles represented to ensure their concerns are recorded in the AD. | | | |
| 4. Questions | 4a. Respondents | 4b. Expected answers | 4c. Criticality |
| 1. State your stakeholder role. List the set of concerns you have that pertain to the architecture whose AD is being reviewed. 2. Find and record all places in the AD where your stakeholder role is listed as being covered. 3. Find and record all places in the AD where your concerns are listed as being addressed. 4. Find and record all places in the framework used (if any) where your stakeholder role is listed as being addressed. 5. Find and record all places in the framework used (if any) where your concerns are listed as being addressed. 6. Record all concerns you have that are <i>not</i> listed as being covered in either the AD or any framework being used or that are listed in an unclear fashion. For each, state the impact of this omission or misunderstanding on project success. 7. For each of your concerns as a stakeholder, find and record the places in the AD where that concern is addressed (not just listed). Explain why you do or do not believe that the concern will be satisfied by the architecture. 8. Find and record the place in the AD that prioritizes the concerns. Explain why you do or do not agree with it. | All stakeholders | All stakeholders should be able to find where in the AD (and framework, if any) their roles and concerns are listed and their concerns are addressed. Every relevant stakeholder and concern should be covered; missing ones should be noted. All concerns should be tied to at least one stakeholder. The architect should provide a convincing argument that the process for identifying stakeholders and their concerns was adequate. In addition to producing satisfactory answers, the respondents should also note the ease or difficulty in using the AD to answer the questions. | Questions revealing missing stakeholders or missing concerns are the most critical. |

| | | | |
|---|-----------|--|--|
| 9. Record important stakeholders that you are aware of that are not listed and whose concerns are not represented in the AD. | | | |
| 10. State how you know that the architecture satisfies the concerns of the missing stakeholders and where this information can be found in the AD. | | | |
| 11. Show where in the AD the generic stakeholders and concerns required by the framework in use (if any) have been listed and addressed. | Architect | | |
| 12. State how you produced the list of stakeholders and their concerns. | | | |
| <p>5. Advice</p> <p>This question set is especially appropriate for an active design review, in which an all-hands meeting is not required. Individual reviewers representing different stakeholder roles and concerns can be engaged separately, perhaps even by telephone or electronic mail, to make sure their concerns are addressed in the AD.</p> <p>By contrast, however, a similar review was carried out as a two-day all-hands workshop for a large U.S. defense project. The first half-day was used to present ISO 42010 terms and approaches. This was a long review because the project is large. Some 30-40 people were involved, and even then some stakeholder communities were overlooked.</p> <p>On a small distance-learning project, a review for this purpose took six hours with a dozen people: six architects and six stakeholders. The agenda devoted two to three hours to the approach and three hours to concerns.</p> | | | |

4.2 Sample Question Set for Reviewing Choice of Framework and Viewpoints

This sample question set assesses the “framework” within which the AD is being developed. By “framework,” we mean something like the (U.S.) Department of Defense Architecture Framework (DoDAF) [DoDAF 2007], or The Open Group Architecture Framework (TOGAF) [TOGAF 1995], as opposed to something like a small-grained object-oriented “application framework.” We are building on the concept of “architecture framework” currently proposed for inclusion in ISO/IEC 42010 WD2, as described in Appendix.

This question set investigates whether the chosen framework, viewpoints related to the framework, and modeling practices are appropriate for use in constructing the AD. If no such framework is in use, some of the questions in this question set might still be used to understand the choice of views (with associated viewpoints or styles, if any) selected for an AD under review.

The questions in this question set do not follow the Active Design Review technique, but rather a conventional question-and-answer approach. They assume the reviewer is familiar with the vocabulary of terms used by the framework (e.g., viewpoints, models, correspondence).

| |
|---|
| 1. Question set name: Reviewing choice of framework and viewpoints |
| 2. Purpose: Use this question set to assess the choice of viewpoints, frameworks, and associated modeling practices to be used in the AD for their suitability for capturing stakeholders' concerns. |

| 3. Stakeholders and concerns: Architects, who specify and then use the frameworks, viewpoints, and modeling practices. Stakeholders who can confirm or refute that the chosen viewpoints and modeling approaches are able to frame their concerns adequately. | | | |
|---|------------------|---|--|
| 4. Questions | 4a. Respondents | 4b. Expected answers | 4c. Criticality |
| <ol style="list-style-type: none"> Do the selected viewpoints and their prescribed models, languages, techniques, evaluation criteria, correspondence rules, and so on, frame the concerns of the stakeholders? Is the framework consistent with the developing organization's required practices and mandated standards? Is the framework consistent with the client's required practices and mandated standards? Does the project have the necessary resources (tools, technologies, methods, and skilled people) to plan and carry out the creation of the AD according to the framework? Is every viewpoint required by the chosen framework(s) included in the AD? Are the concerns that are covered by the framework well-aligned with the concerns of the stakeholders? Does the framework include concerns that are not concerns of your stakeholders? Do the viewpoints frame the stakeholder concerns? For each viewpoint, are its models clear and well-defined? Do the models provide enough information for determining whether the concerns framed by the viewpoint have been satisfied? For each model, are there appropriate tools, notations, experience/training, documentation, and techniques in place within the architecture team for applying the model? What correspondences exist between models in the same viewpoint or across different viewpoints? Which of these correspondences came from the framework and which came from the architect's own selection of viewpoints? Which concerns are addressed by each correspondence, to the extent that the correspondence provides enough information to let us determine whether the concern has | All stakeholders | <p>"Yes" is generally the expected answer. Respondents should be prepared to point out where material in the AD supports their answers.</p> | <p>Questions are most critical that address areas of high risk:</p> <ul style="list-style-type: none"> architectural viewpoints have not been selected architectural viewpoints not well-defined stakeholder concerns that cannot be captured using the representational resources of the selected viewpoints viewpoints that are not achievable due to resource or tool constraints |

| | | | |
|---|--|--|--|
| been satisfied? | | | |
| 12. Are all concerns addressed either by one or more models or by one or more correspondences among models? | | | |
| 13. Is there a smaller set of viewpoints, models, and correspondences that would also cover all of the stakeholder concerns? | | | |
| 14. Is it feasible that the views drawing upon these models, viewpoints, and framework(s), can be constructed with the available tools, techniques, and people, within the time and funding available? | | | |
| 15. Is there rationale captured for the choice of framework, viewpoints, models, and correspondences? | | | |
| <p>5. Advice</p> <p>Experience with SEI ATAM-based evaluation exercises, SEI Quality Attribute Workshops (QAWs), and other assessments suggests that looking at stakeholders, concerns, and viewpoints provides a high return on investment (ROI). When performed early, a framework review focuses the architectural work; and when done later, it helps bring misguided/off-track efforts back on-track by focusing effort on the most critical concerns and viewpoints/models.</p> <p>A framework review can be undertaken before the AD and its views are completed, or even begun, to determine how useful selected viewpoints and modeling resources are in framing stakeholders' concerns. Because constructing and modeling views can be time consuming and expensive, it is practical to ascertain that you have the right questions, before spending significant resources answering them.</p> <p>A framework review may be a good antidote to organizations that have a one-size-fits-all ("we always use these six viewpoints!") approach to architecture documentation to detect mismatches between the models the architect is expending effort on and the prevailing architectural concerns for the system.</p> <p>It is also very helpful to capture (a priori) the evaluation criteria for models, viewpoints, or frameworks. Again that's a good thing to know before you spend a lot of time working with a representation. Architecture products captured in PowerPoint may be effective as communications vehicles but are not very analyzable.</p> | | | |

4.3 Sample Question Set for Supporting Evaluation

When an architecture is subjected to a comprehensive evaluation, the AD is the vehicle for communicating the architecture to the reviewers, or for at least substantiating the architect's presentation of the architecture. Therefore, it is useful to review the AD before an architecture evaluation takes place to see if it contains the necessary information to allow the evaluation to go forward. By extension, such a review determines whether the architecture is ready (complete enough) to be evaluated.

| |
|---|
| 1. Question set name: Supporting evaluation |
| 2. Purpose: Use this question set to determine whether the architecture is ready to be evaluated. This helps ascertain whether evaluation stakeholders have sufficient information to do their job and know when their job is completed. The emphasis is on the artifacts needed for analysis. |
| 3. Stakeholders and concerns: The business manager is the spokesperson for the business goals the system is |

meant to support. These goals include what the customer wants to build and the objectives of the organization building the system. The business manager is concerned with how the technical solution supports the business goals. Often the project software manager fulfills this role for the purpose of the evaluation.

The architect is concerned with whether the AD supplies sufficient information for analysis and how usable the AD is in supporting an evaluation. The architect would like to use the AD to determine the implications of a design decision in terms of technical considerations, difficulty, and risk.

The team preparing to conduct an architecture evaluation is concerned with knowing what to evaluate, on what basis, and whether the AD supplies sufficient information for analysis.

| 4. Questions | 4a. Respondents | 4b. Expected answers | 4c. Criticality |
|--|------------------|---|---|
| <ol style="list-style-type: none"> Are the business goals the system must satisfy clearly articulated and prioritized? Is it clear how the business goals determine the requirements? Is there a mapping between business goals and requirements? Are the requirements prioritized according to business importance? Is there traceability between the business goals and the technical solution? That is, can you navigate from business goals to architecturally significant requirements (ASRs), to technical decisions and associated risks, and finally back to implications on achieving the business goals? What criteria are used to determine whether the architecture is supporting the business goals? How might the system change over its lifetime of deployment (including retiring the system)? | Business manager | The business manager and the architect should provide a convincing argument that the documentation captures the important analysis artifacts that allow one to navigate from business goals to architecturally significant requirements, to technical decisions and associated risks, and finally back to the implications of these requirements, decisions, and risks on achieving the business goals. | Questions revealing missing analysis artifacts (e.g., architecturally significant requirements, architecture decisions) are the most critical. Questions indicating incompleteness or ambiguity in conducting the analysis are also critical. |
| <ol style="list-style-type: none"> Is the context of the system (or subsystem) clearly defined? Have the stakeholders and their concerns been clearly defined? Have the requirements, constraints, standards, and quality assurance policies been clearly defined? Are the ASRs that the system must satisfy clearly articulated and prioritized according to their impact on the architecture? Are the ASRs clear and unambiguous? Are they "testable"? Have they been prioritized? Is it clear which techniques the architect used to satisfy the ASRs? Have alternatives that were considered but not chosen been documented? Is it clear how the architecture fulfills the other requirements that are not ASRs? Have you identified the key decisions? Have you captured the rationale for your key decisions? | Architect | | |

| | | | |
|---|--|--|--|
| <p>15. Can you describe the runtime resources consumed for each concern that affects the operation of the system?</p> <p>16. Can you describe the change impact (estimated size/difficulty of the change) for those modifiability concerns that lead to changed design elements?</p> <p>17. Can you determine the views necessary to analyze each ASR? Does the AD provide the views necessary to cover the ASRs?</p> <p>18. Within each view, are its models clear? Are its models well-defined by the viewpoint? Do the models address the ASRs? Which ASRs are addressed by the models in this view (to the extent that the model provides enough information for determining whether the ASRs have been satisfied)?</p> <p>19. Are all ASRs addressed by either one or more models or one or more correspondences among models?</p> <p>20. Have you done any preliminary analysis? Have these results (including architectural issues and risks) been articulated?</p> <p>21. How will the architecture be introduced and retired within the business?</p> <p>22. Is the current document complete in the sense that all the information is documented? If not, are there placeholders for what has yet to be documented along with descriptions of what still needs to be worked out?</p> <p>23. Can you navigate through the material during the evaluation to show the decisions made to address stakeholders' concerns?</p> | | | |
| <p>5. Advice</p> <p>Depending on the scope of the evaluation, there could be some overlap with the “Question set for supporting development.” Analysis could include “buildability” or “feasibility in building the system as the customer describes it.” There is no overlap when evaluation is more narrowly scoped in the sense of identifying decision points and the rationale for selecting alternatives. In this case, the AD is treated as a sketch that shows alternatives rather than a blueprint from which to build the system.</p> <p>If the AD uses frameworks and viewpoints, the “Question set for reviewing choice of framework and viewpoints” could be answered in conjunction with this review. If the AD does not use these concepts explicitly, some of the questions could still be used to understand the documentation.</p> <p>The business manager and the architect share their answers to the questions with the evaluation team. The evaluation team may answer the questions separately to varying degrees of detail in order to validate the results.</p> <p>The set of questions will be tailored according to the scope and objectives of the evaluation (any combination of the system, stakeholders, ASRs, views, and decisions).</p> | | | |

4.4 Sample Question Set for Supporting Development

Architecture has value by driving a conforming implementation—that is, that the developers can follow the specifications and constraints of the architecture. The purpose of conducting a review

for supporting development is to determine whether there is enough information in the architecture for the development stakeholders to do their jobs. A closely related task is to determine if the AD is sufficient to determine whether a system's implementation actually conforms to the architecture described in the AD. The emphasis there is on the ability of the AD to identify *conformance points* for the implemented system, with the expectation that a subsequent review or audit (possibly using automated tools) will actually determine conformance of the system to the architecture (described by the AD).

| <p>1. Question set name: Supporting development</p> | | | |
|--|------------------|--|--|
| <p>2. Purpose: Use this question set to determine whether the AD contains enough information to “drive” a conforming implementation. This helps ascertain whether development stakeholders have sufficient information to do their job and know when their job is completed. The focus is less on analysis and more on comprehension and completeness of the AD.</p> | | | |
| <p>3. Stakeholders and concerns</p> <p>Architects are concerned with whether their AD is ready to pass to developers.</p> <p>Designers and implementers are concerned with knowing what to build—that is, what they must do in order to implement the architecture.</p> <p>Software managers are concerned with estimating and/or predicting needed development resources (budget, schedule).</p> <p>Developers are concerned with when to enter test.</p> <p>Testers are concerned with whether the AD supplies sufficient information to enable architecture-based testing and to determine when to exit test.</p> <p>QA stakeholders are concerned with whether the AD supplies sufficient information to enable quality assurance and to make clear when QA is complete. A special kind of QA stakeholder is the “conformance checker,” who is concerned with how to tell whether an implementation conforms to the architecture.</p> <p>Integrators are concerned with whether the AD supplies sufficient information to plan integration.</p> <p>Fielders are concerned with whether the AD supplies sufficient information to plan deployment.</p> <p>Customers and program managers have indirect concerns about whether the AD is usable by developers and how the architecture is constrained by existing components.</p> | | | |
| 4. Questions | 4a. Respondents | 4b. Expected answers | 4c. Criticality |
| <p>1. Can you identify the full set of implementation units (element to be implemented)?</p> <p>2. Can you determine which units require development (and integration and test) resources?</p> <p>3. For each unit requiring development, can you make predictions in terms of use of development resources, variance, and risk?</p> <p>4. Can you determine development dependencies between implementation units?</p> <p>5. Can you identify runtime dependencies between units?</p> <p>6. Can you lay out a schedule for this development?</p> <p>7. Can you lay out a schedule for an architectural prototype?</p> <p>8. Can you tell if you have enough</p> | Software manager | <p>In all cases, the stakeholders should provide a convincing argument that the documentation captures the important artifacts that allow one to implement the architecture.</p> <p>In addition to producing satisfactory answers, the respondents should note the ease or difficulty in using the AD to answer the questions.</p> | <p>Questions revealing incompleteness or misunderstanding of artifacts are the most critical. In this case, the AD is treated as a blueprint from which to build the system or to which the built system must conform.</p> |

| | | | |
|--|---|--|--|
| development resources? | | | |
| 9. Does the AD overconstrain the stakeholders (e.g., developers, integrators)? | | | |
| 10. Does the AD identify opportunities for parallel development? Can you identify units that can be implemented in parallel? | | | |
| 11. Can you identify the allowed and prohibited dependencies between implementation units? | Designers and implementers (including unit testers) | | |
| 12. Can you identify applicable architectural constraints, rules, principles, styles, patterns, etc. on units or their aggregation? | | | |
| 13. Can you navigate from an implementation unit to its associated requirements (formal, derived, quality, performance, and design constraints)? | | | |
| 14. Can you determine a test approach for the set of implementation units? | | | |
| 15. Can you determine approaches for error handling, resource management, human-computer interaction, data management and persistence, variation and variability (e.g., across a product line or evolution over time), etc.? | | | |
| 16. Can you determine what is likely to change and how it impacts your design? | | | |
| 17. Can you tell how solid each decision is? | | | |
| 18. Can you tell what needs to change as the result of entering a new cycle? | | | |
| 19. Do you understand how conformance to the AD will be determined? | | | |
| 20. Does the AD identify opportunities for parallel development? Can you identify units that can be implemented in parallel? | | | |
| 21. Can you identify what units must be integrated? | Integrators and fielders | | |
| 22. Can you determine the resources needed to run the unit? | | | |
| 23. Can you determine the integration test obligations? | | | |
| 24. Can you identify runtime (e.g., | | | |

| | | | |
|--|---|--|--|
| load, elaboration) dependencies between units? | | | |
| 25. Do you understand how conformance to the AD will be determined? | | | |
| 26. Can you determine which units can be cost-effectively tested in isolation? | Testers (not unit testing, but rather architecture-based testing) | | |
| 27. For each unit, can you determine what is needed (e.g., data, special hardware, other units) to test it? | | | |
| 28. For each unit, can you determine what constitutes test success criteria? | | | |
| 29. Can you test the system as a whole? | | | |
| 30. Is the AD baselined? | QA stakeholders | | |
| 31. Is there a history of changes to the AD? | | | |
| 32. Does the AD identify key decisions? | | | |
| 33. Does the AD capture the key decisions and design rationale? | | | |
| 34. Does the AD articulate “open decisions” deferred to implementation? | | | |
| 35. Are inconsistencies known and documented? | | | |
| 36. Are there known associations between each view’s models and developed/delivered artifacts? (e.g., if we have a “deployment view” in the architecture, do we have a “packing list” for the system?) | | | |
| 37. Are specific conformance points identified in the views? For each such point, do we know which view and model captures this information and which artifact/artifacts must conform? Is there a documented method for checking conformance (e.g., inspection, developer test, formal qualification test) | | | |
| 38. What questions, concerns, or issues have the developers raised during their work? How are these captured/resolved in the AD? How has the AD changed in response to these concerns? | | | |
| 39. Are the test approaches and artifacts consistent with the AD? (Could include precise trace or an | | | |

| | | | |
|--|------------------|--|--|
| <p>informal assessment. This is particularly associated with “use case” kinds of views, where you want the testers to test the known use cases that the architecture should have addressed.)</p> <p>40. Is there a process for establishing conformance?</p> <p>41. Does the content of the AD support this process?</p> | | | |
| <p>42. Can you identify open, partially resolved, or unresolved issues in the AD?</p> <p>43. Can you identify where automated tools will be used? Does the AD have the right content that is in a format that can be processed by the tools?</p> | All stakeholders | | |
| <p>5. Advice</p> <p>This question set might overlap with a question set that reviews the AD for its ability to support an architecture evaluation, in that the evaluation could analyze for “buildability” or “feasibility in building the system as the customer describes it,” which are, of course, among the developer concerns addressed here.</p> <p>A subset of the question set may be used in a more specialized review for supporting planning.</p> | | | |

4.5 Sample Question Set for Identifying Architecturally Significant Requirements and Key Design Decisions

Every architecture is the result of making key design decisions that satisfy *architecturally significant requirements*. These are high-priority requirements that cause the architecture to look much different than it otherwise would have. For example, a requirement for 24/7 availability often leads to introduction of hardware, software, and/or data redundancy, as well as failure-detection and -recovery mechanisms. None of these would be present in the absence of such a requirement. An example of a key design decision is the adoption of a particular architectural style [Clements 2003], the introduction of specific tactics [Bass 2003], or the imposition of design constraints for designers downstream of the architect. These decisions are made specifically to address architecturally significant requirements. The purpose of this question set is to make sure the AD addresses the architecturally significant requirements, as well as identifying the key design decisions made to satisfy each one, and to make sure that none of them have been overlooked.

| |
|--|
| <p>1. Question set name: Identifying architecturally significant requirements and key design decisions</p> |
| <p>2. Purpose: Use this question set to identify and vet key decisions that should be captured by the architecture. Such decisions should drive the subsequent development and review of the architecture towards those architectural decisions that have the most significant impact. Often, such decisions are motivated by architecturally significant requirements, particularly quality attribute requirements such as throughput or availability requirements for the system as a whole.</p> |
| <p>3. Stakeholders and concerns</p> <p>Architects are concerned with having the right set of viewpoints to capture architectural decisions that show how the architecture (and the system as built) will satisfy these requirements, as well as having the right set of viewpoints to properly frame the design solution (in particular, without overconstraining the design).</p> <p>Acquirers are concerned with assuring that the AD and the architecture it describes reflect their driving requirements and that the key design decisions made are feasible and buildable.</p> |

| <p>Analysts are concerned with making sure the contents of the architectural views tie back to the driving architectural decisions in order to understand how a specific decision impacts the architecture's ability to satisfy its requirements. They are also concerned with making sure the contents of the view help frame the cost, schedule, and implementation effort to implement the decision and with being able to tell which requirements have the most potential impact on the architecture, so a trade-space for implementing that set of requirements can be constructed.</p> | | | |
|---|-----------------|---|--|
| 4. Questions | 4a. Respondents | 4b. Expected answers | 4c. Criticality |
| <ol style="list-style-type: none"> Are specific architecturally significant requirements (i.e., the subset of functional, quality attribute, and business requirements that "shape" the architecture under consideration) identified? Are ASRs represented in a clear, unambiguous manner (c.f., 6-part quality attribute scenarios [Bass 2003])? Is the utility of the requirements documented in terms of what the system does and how it meets the customer's expectations? Are there remaining requirements that could come up later and have a significant impact on the architecture? How will the architecture (and the architecting process) react to the emergence of new ASRs? Is the relationship between ASRs documented and understood (e.g., between performance requirements in a distributed system and the bandwidth, reliability, and stability of the supporting network transport systems)? Are decisions represented in a clear, unambiguous manner (c.f., architectural tactics [Bass 2003])? Is the rationale for key decisions captured? Are the costs and resources associated with implementing the decisions documented? Are there remaining architectural decisions or impacts (e.g., issues or problems that could come up during deployment, deferred decisions that need to be bound later)? | Architect | Positive answers are expected, as well as the ability of the participants to point out specific places in the AD to justify their positive answers. | <p>The AD will need to have some way to identify and refer to architecturally significant requirements (ASRs).</p> <p>The AD needs to have mechanisms that capture the relevant aspects of ASRs and the architectural decisions that trace to specific ASRs.</p> <p>The AD needs representations and models that capture specific decisions.</p> <p>The AD should have a means to capture the rationale for these decisions.</p> |

| | | | |
|---|--------------------------------|--|--|
| <p>7. Is there a mapping between decisions and requirements?</p> <p>8. Is there a process or method for establishing and maintaining traceability from business goals, requirements, and decisions? Is there a process for conducting trade studies to balance competing requirements, designs, or implementation approaches? If so, does the AD contain the correct data to support these processes?</p> <p>9. If the architecture is part of a life cycle or process that includes a procurement decision, does the AD contain the appropriate information to support the procurement process?</p> <p>10. Do the customers/acquirers have the right information to understand the key decision and how that decision meets the system requirements and constrains the design and implementation of the system?</p> | <p>Acquirers and analysts</p> | | |
| <p>11. Are specific driving architectural decisions identified? Is the relationship between them documented and understood (e.g., between performance requirements in a distributed system and the bandwidth, reliability, and stability of the supporting network transport systems)?</p> <p>12. Are decisions represented in a clear, unambiguous manner? Is the rationale for key decisions captured? Are the costs and resources associated with implementing the decisions documented?</p> <p>13. Are there remaining architectural decisions or impacts (e.g., issues or problems that could come up during deployment, deferred decisions that need to be bound later)?</p> <p>14. Do you understand how the AD will identify constraints and implementation responsibilities (e.g., delegated decisions)?</p> | <p>All listed stakeholders</p> | | |

5. Advice

This can be accomplished via a two-phased approach: (1) Architects present their understanding of the ASRs and obtain validation from customers and analysts. (2) Architects present the architectural decisions based on the ASRs to both customers and analysts and to implementers. Customers and analysts should come away believing the architecture is positioned to meet the ASRs, and developers should understand how these decisions constrain and drive the subsequent system design and implementation.

4.6 Sample Question Set for Reviewing for Conformance to ISO/IEC 42010

This review assesses whether the AD conforms to the requirements of ISO/IEC 42010, *Systems and Software Engineering—Architecture Description*,² summarized in Appendix.

In the following question set, clause numbers refer to ISO/IEC Working Draft 4 42010 [ISO/IEC WD4 42010:2009].

| | | | |
|--|------------------------|---|---|
| 1. Question set name: Reviewing for conformance to ISO/IEC WD4 42010 | | | |
| 2. Purpose: Use this question set to assess the conformance of the AD to the requirements of the draft international standard ISO/IEC WD4 42010. Conformance to the standard may be a prerequisite to acceptance of the AD as a deliverable or to other reviews. | | | |
| 3. Stakeholders and concerns: Architects, acquirers, and analyst all have the following concern: Does my AD meet all of the conformance points of the standard? Can conformance be verified? | | | |
| 4. Questions | 4a. Respondents | 4b. Expected answers | 4c. Criticality |
| <ol style="list-style-type: none">Does the AD contain the appropriate administrative data (5.2, date of issue, version status, issuing organization, change history, summary, scope, context, glossary, and references)?Is architectural documentation required by the organization included in the AD?Who are the specific stakeholders for this AD? Is there evidence the architect has given consideration to these stakeholder classes: Users of the system, system acquirers, system developers, and system maintainers?Are the stakeholders' concerns captured? (5.3) Does the AD show evidence of having considered the purpose of the system's missions; the appropriateness of the system for use in fulfilling its missions; the feasibility of constructing the system; the risks of systems development; the risks of operation to users, acquirers, and developers; maintainability; and deployability and evolvability? | Architects | Positive answers are expected, as well as the ability of the participants to point out specific places in the AD to justify their positive answers. | For the purpose of ascertaining conformance, all requirements in ISO/IEC 42010-2007 are of equal importance, and all are mandatory. (There are no tailoring options in the standard.) |

² As noted earlier, ISO/IEC 42010:2007 is the ISO adoption of ANSI/IEEE 1471-2000, and is identical to that earlier standard. As this note is being written, a joint revision of ISO/IEC 42010 and ANSI/IEEE 1471 is ongoing. The question set below contains questions that address the expected future form and content of the ISO/IEC 42010 revision, including new topics such as architecture frameworks and model correspondences, based on author participation in the standards bodies. Should the revision take a different direction, then this question set should be modified accordingly. The remainder of the question set is valid for IEEE 1471-2000 and ISO/IEC 42010:2007.

| | | | |
|--|------------------------|--|--|
| <p>5. Is every stakeholder and every concern covered by at least one viewpoint?</p> <p>6. Is each viewpoint identified? Is there a definition for each viewpoint used in the AD? Does each viewpoint definition include: viewpoint name; identification of the stakeholders addressed by that viewpoint; the architectural concerns framed by that viewpoint; and the languages, modeling techniques, and analytical methods used with views of that type?</p> <p>7. If the viewpoint comes from an external source, is it fully defined and identified in that source? Is there an association between that viewpoint and the stakeholders' concerns? Are models/modeling techniques identified? Does the viewpoint contain analysis techniques, rules, or constraints (5.4)?</p> <p>8. Is there a view for each viewpoint? Does the view correctly use/implement the models required by its viewpoint? Does the view cover the system under review (5.5)? Is the view-viewpoint relationship 1-to-1?</p> <p>9. Does each view contain: an identifier, introductory information, configuration information as defined by the using organization, and one or more models?</p> <p>10. Are there model correspondence rules? For each such rule, is there a model correspondence satisfying each rule (5.7)?</p> <p>11. Does the AD cite an existing architecture framework? Is each viewpoint in the framework used in the AD? Does the AD capture all of the framework's model correspondence rules (6.1)?</p> <p>12. Are any known inconsistencies between views documented?</p> <p>13. Does the AD contain the rationale for its architectural decisions, such as</p> <ul style="list-style-type: none"> • Selection of viewpoints and models/modeling techniques? • Viewpoint Correspondence Rules? • Key decisions captured within each view? • Known inconsistencies (5.7)? | | | |
| <p>14. Is the set of stakeholders and concerns complete (5.3)?</p> <p>15. Is the set of viewpoints both complete and minimal (5.4)?</p> <p>16. Is the set of correspondence rules (if used) appropriate (5.7)?</p> <p>17. Are the views complete? Do they communicate the key decisions (5.5)?</p> <p>18. Is the set of correspondences complete (5.7)?</p> | Acquirers and analysts | | |

| | | | |
|---|--|--|--|
| 19. Does the rationale capture sufficient information to assist reviewers and analysts in understanding the architecture and its decisions (5.8)? | | | |
| 20. Do the set of viewpoints and/or the selected architecture framework match contractual requirements and/or institutional practices (6.1)? | | | |
| <p>5. Advice</p> <p>“Complete” here is expected to be a value judgment in the review, rather than any formally determined property. The stakeholders need to understand the context (including resource constraints) as part of evaluating “completeness.” Generally, “complete” should be interpreted as “good enough to meet our expectations for this system within the context in which we are developing it.” These rules should not be required as having the architecture description account for every (software equivalent of a) nail in the structure.</p> <p>Each item chosen above directly maps to conformance points in ISO/IEC 42010-2007. The terms in this section are taken directly from ISO/IEC WD4 42010:2007, clauses 3 and 4.</p> | | | |

5 Examples of Constructing a Review

This section shows examples of approaches for constructing reviews from the question sets in response to stakeholders' needs. First, we show detailed examples of using the structured approach to describe two reviews. Next, we show how the question sets can be used in a wide range of circumstances.

5.1 An Example: AD Reviews in the Context of the ATAM

The *ATAM Reference Guide* describes guidelines for evaluating the readiness of an organization to proceed with an architecture evaluation as shown in Figure 3. Among the guidelines are criteria for reviewing the AD.

A responsible architect has been identified and is committed to participating in the ATAM evaluation.

A business manager or system sponsor has been identified and is committed to participating in the ATAM evaluation.

The architectural documentation has been provided to the evaluation team, and, according to the ATAM evaluation team, it is sufficient (not too small, not too large) to proceed with the ATAM evaluation.

The business drivers and architectural presentation have been provided to the ATAM evaluation team. The presentations are sufficient but well within the 1.5-hour time constraint for each presentation during the ATAM evaluation.

System stakeholders have been identified and are committed to participating in the ATAM evaluation. The list of participants and their roles with respect to the system has been sent to the ATAM evaluation team.

Figure 3: ATAM Phase 0 Go/No-Go Criteria

In this section, we show how the criteria for reviewing the documentation could be described in terms of the approach for reviewing architecture documentation. We follow the steps that constitute a methodical architecture document review from Section 3.

- **Step 1: Establish the purpose of the review.** The ATAM Phase 0 readiness review is an example of reviewing to see if the AD is suitable for supporting architecture evaluation or analysis (Section 2.1). The purpose of conducting an ATAM evaluation is to assess the consequence of architectural decisions in light of quality attribute requirements and business goals, so the purpose of the AD review will be to ensure that those analysis artifacts (architectural decisions, quality attributes, and business goals) are documented.
- **Step 2: Establish the subject of the review.** The ATAM requires the customer to make a presentation that describes the candidate system with sufficient details to convey the main architecturally significant requirements. After that presentation, the customer and the evaluation team discuss and agree on the necessary architecture documentation and artifacts to be reviewed, which include stakeholders and concerns, how the documentation is laid out to serve stakeholders (roadmap and view template), additional information about what the architecture is (system overview, mapping between views, directory, and glossary), and why the architecture is the way it is (system background, design constraints, and rationale).
- **Step 3: Build or adapt the appropriate question set(s).**
The ATAM Phase 0 go/no-go criteria for reviewing the architecture documentation from

Figure 3 leads to the selection of the question sets for capturing the right stakeholders and concerns and supporting evaluation (see Table 2). If a framework is used, the question set for reviewing the choice of the framework and associated viewpoints is included as well.

Table 2: Building an ATAM AD Review from Question Sets

| ATAM Criteria | Associated Question Set(s) |
|--|---|
| A responsible architect has been identified ... | capturing the right stakeholders and concerns |
| A business manager or system sponsor has been identified ... | capturing the right stakeholders and concerns |
| The architectural documentation has been provided ... | supporting evaluation reviewing choice of framework and viewpoints |
| The business drivers and architectural presentation have been provided ... | supporting evaluation |
| System stakeholders have been identified ... | capturing the right stakeholders and concerns |

Typically, one person represents the review team during the customer presentation of the candidate system. During this time, the questions from the question sets can be asked of the business manager and architect. Typically after the presentation, all the evaluation team members spend a limited amount of time reviewing the documentation separately and then meet as a team to apply the go/no-go criteria. The question sets are used as a checklist by the evaluators when reading the document, since the stakeholders are not available for questioning (although important questions could be asked of the business manager or the architect if necessary to clarify an issue).

- **Step 4: Plan the details of the review.** The ATAM readiness review is conducted during Phase 0 of an ATAM evaluation. The scheduling of the ATAM evaluation determines when in the life cycle the readiness review is conducted. The ATAM may be used throughout the life cycle when there is a software architecture to evaluate. It can be used after an architecture has been specified but there is little or no code, or it can be used to evaluate the architecture of an existing system. The readiness review is conducted after the client representative describes the candidate system and delivers architecture documentation to the evaluation team. The evaluation team meets separately to apply the go/no-go decision criteria to the candidate system.
- **Step 5: Perform review.** The evaluation team meets separately to apply the go/no-go criteria when reviewing the AD. The degree to which the documentation meets the criteria is noted so that feedback can be offered to the client and the process for the evaluation can be refined. The question sets are tailored and used in the following ways:
 - capturing the right stakeholders and concerns. The question set is used in an auditing function. The reviewers check the AD to make sure that the following are documented: a list of the stakeholders' roles and concerns, the criteria the architect used to produce that list, and how the architecture satisfies the concerns.
 - reviewing the choice of framework and viewpoints. If the AD uses frameworks and viewpoints, the review team uses the question set for the "Reviewing choice of framework and viewpoints" in conjunction with the other questions. If the AD does not use

these concepts explicitly, some of the questions could still be used to aid understanding of the documentation.

- supporting evaluation. The review team applies the questions against the AD (business presentations, architecture presentation, and architectural documentation) rather than asking the business manager and architect the questions directly, given the limited time and resources of Phase 0. If the AD fails to provide the expected answers, feedback is given to the customer to improve the AD.
- **Step 6: Analyze and summarize results.** Judgment is rendered in the form of a go or no-go decision. If a go decision is made, the knowledge from the review is used to refine the list of appropriate stakeholders, determine the expertise needed for the evaluation team, and make suggestions for improving the architecture documentation. If a no-go decision is made, the knowledge of the review is used to explain to the client the reason for declining the work and suggestions for remediation steps to enable future work.

5.2 An Example: AD Reviews in the Context of SARA

The Software Architecture Review and Assessment (SARA) report documents a framework for software architecture reviews [SARA 2002]. This framework was developed during 1999–2001 to “provide concrete, practical, experience-based guidance on how to conduct architectural reviews ...represent[ing] the collected best practices of a wide group of industrial architects and consultants” (Unless otherwise noted, all quotations are taken from the SARA framework. [SARA 2002].)

The SARA framework addresses

- a conceptual model of software architecture and reviews
- a generic process and workflow with which to execute an architecture review or assessment
- definitions of typical activities that may take place in a review
- a catalogue of methods and techniques that can be used to support the evaluation; and a means for documenting and incorporating new review methods into the workflow
- pragmatic issues of architecture reviews
- case studies
- documentation templates to use

In this section, we show how the approach for reviewing architecture documentation described in this note can be used within an architecture assessment carried out in the context of the SARA framework.

SARA recognizes that there are various kinds of objectives one may have for an architecture review:

1. particular purposes such as
 - a. certifying the conformance to some standard
 - b. assessing the quality of the architecture
 - c. identifying opportunities for improvement

- d. improving communication between stakeholders.
2. review objectives “defined by the life-cycle stage of the project.”
3. review objectives focused on specific aspects of an architecture.

These aspects can include the

- fit of the architecture to the problem or mission statement,*
- partitioning of system responsibilities to subsystems and components,*
- specific qualities (i.e., scalability, performance, etc.) to be architecturally controlled*
- partitioning of the architectural design responsibilities,*
- identification of skills to implement the system,*
- verification of scenarios representing the critical functionality of the system, and*
- overall feasibility and specific risks of the architecture.*

A number of these objectives (but not all) are amenable to analysis with the approach to reviewing architecture documentation. Some have already been addressed in the question set examples in Section 4.

SARA recognizes that it is not a closed, complete framework.

To efficiently and completely address each step of an architecture review may require a specialized method. Each method is defined by: a set of steps (a process), an associated analytic technique, a notation, a set of outputs (work products), and a set of roles for the participants. These are ideally associated with estimates of cost and time.

To this end, the SARA report provides a template for describing new methods that can be used with others to support an evaluation. Table 3 shows how the approach would be recorded using the SARA method template and added to the SARA body of knowledge.

Table 3: Reviewing Architecture Documentation as a SARA Method

| | |
|--------------------|---|
| Name | <i>A succinct name given to the method or technique:</i> Reviewing Architecture Documentation (RAD) |
| Context | <i>In which circumstances would you invoke (execute, use) the technique; which activities of the architecture review workflow does it support? What problem does it help solve?</i> RAD requires availability of the AD. It can be used for determine fitness for purpose of the AD, for a wide set of purposes. |
| Purpose | <i>What does the technique achieve? What additional insight does it provide? What intermediary artifact does it produce?</i> RAD produces review purposes, identification of involved stakeholders and concerns, and a question set to be used in each review. (See Question Set template, Section 4, Figure 2.) |
| Input | <i>What are the artifacts that the technique uses?</i> The principal input is the AD. |
| Output | <i>What are the results of applying the technique? What artifacts does it produce or update, and how do you interpret these results?</i> Responses to the question set are captured and analyzed against the expected answers. |
| Steps | <i>What is the series of steps or workflow for this method (if it is a complex technique)?</i> Step 1: Establish the purpose of the review. Step 2: Establish the subject of the review. Step 3: Build or adapt the appropriate question set(s). Step 4: Plan the details of the review. Step 5: Perform the review. Step 6: Analyze and summarize the results. |
| Roles | <i>Who are the participants?</i> The AD reviewers and other stakeholders, depending on the specific purpose of the review. |
| Estimates | <i>What is the estimated effort to apply the technique?</i> The effort to apply RAD varies with the AD, the purpose, and so on But most question sets can be answered in a single session. |
| Reference | <i>Where has this technique been published/described?</i> Nord, R. L.; Clements, P. C.; Hilliard, R.; & Emery, D. <i>A Structured Approach for Reviewing Architecture Documentation</i> (CMU/SEI-2009-TN-030). Software Engineering Institute, Carnegie Mellon University, 2009. |
| Tools | <i>What tools support this technique?</i> None |
| Alternative | <i>What other technique could be used for a similar purpose?</i> None |

5.3 Building Reviews from Question Sets

Table 4 shows how the sample AD question sets defined in Section 4 can be applied for various review purposes, consistent with those given in Section 2.1. The left-hand column is derived from ISO/IEC 42010, clause 4.4, “Uses of architectural descriptions.” The right-hand column suggests the question sets that might be included in an AD review to be used to support the stated purpose. Of course, AD reviews are not appropriate to answer all of the architectural questions being posed and are often used in conjunction with other assessment techniques. For example, using the architectural description to facilitate communications between acquirers and developers as a part of contract negotiations and bid assessment, a new technology may be proposed; evaluating the maturity or usability of the proposed technology is difficult or impossible to do with only a description. However, it may be readily addressed via a prototype.

The life-cycle stages (described in Figure 1 from Section 2.4) most often associated with the use are also shown.

Table 4: Architecture-Related Reviews and Their Associated Question Sets

| Uses of Architectural Descriptions | Associated Question Set(s) |
|---|---|
| Analysis of alternative architectures Stage: Concept | capturing the right stakeholders and concerns |
| Business planning for transition from a legacy architecture to a new architecture Stages: Concept, Development, Utilization | capturing the right stakeholders and concerns |
| Communications among organizations involved in the development, production, fielding, operation, and maintenance of a system Stages: Development, Production, Utilization, Support | supporting development |
| Communications between acquirers and developers as a part of contract negotiations Stages: Development, Production | capturing the right stakeholders and concerns (Before sending out the documentation, the acquirer wants to know “Is this what I want to have built?”) supporting development (Upon receiving the documentation, the developer wants to know “Can I build this?”) |
| Criteria for certifying conformance of implementations to the architecture Stage: Development | supporting development (concentrating on checking the conformance of the implementation) |
| Development and maintenance documentation, including material for reuse repositories and training materials Stage: Development, Support | supporting development |
| Input to subsequent system design and development activities Stage: Development | supporting development (may also include identifying architectural drivers where subsequent design is focused on addressing the technical risks associated with them) |
| Input to system generation and analysis tools Stages: Development, Production | supporting development (specialized case to determine if the AD is ready to be used for development although there are no people involved. The AD still needs to be reviewed to see whether it has the right content and whether the relevant content is in a format that can be processed by the tools.) |
| Operational and infrastructure support; configuration management and repair; redesign and maintenance of systems, subsystems, and components Stages: Utilization, Support | capturing the right stakeholders and concerns supporting development (concentrating on checking for conformance) |
| Planning and budget support Stages: Utilization, Support | supporting development |
| Preparation of acquisition documents (e.g., requests for proposal and statements of work) Stages: Concept, Development | supporting evaluation supporting development |
| Review, analysis, and evaluation of the system across the life cycle Stages: all | supporting evaluation |
| Specification for a group of systems sharing a common set of features (e.g., product lines) Stage: Concept | capturing the right stakeholders and concerns (with emphasis on finding commonality and variations) supporting development |

6 Related Work

A number of frameworks have been proposed that compare evaluation methods to help users better understand which method to choose and its applicability to a particular situation [Babar 2004, Dobrica 2002]. There is at least one attempt in the literature to provide a comprehensive framework for architecture reviews and assessment [SARA 2002]. These frameworks use different criteria to compare the methods, but the criteria generally fit within four perspectives:

1. Context
 - goals of the method
 - scope in terms of stakeholders, quality attributes, architecture views
 - life-cycle phase
 - application domain
2. Content
 - inputs required and form of software architecture documentation recommended
 - outputs produced
 - activities performed to achieve goals
3. Mechanisms
 - stakeholder involvement
 - techniques for determining architecturally significant requirements and performing analysis
 - tool support (document processing, architecture representation, and analysis)
4. Validity
 - validation of the method's output
 - repeatability of the method
 - maturity of the method

While the frameworks are meant to help differentiate methods for evaluating architecture design, much of the criteria are still applicable and illuminating when one is trying to understand what goes into reviewing architecture documentation.

Hämäläinen and Markkula have proposed what they call a “question framework” for assessing the quality of AD [Hämäläinen 2007]. It consists of four separate questionnaires addressing (1) stakeholders and purpose, (2) content, (3) presentation and visualization, and (4) documentation management. Examples of questions include

Are the stakeholders of the documentation defined? If so, who are they?

Is the purpose of the documentation in relation to these stakeholders defined? If so, what is it?

Does the documentation present the needs and concerns of stakeholders correctly?

Does the information in the documentation reflect the current enterprise?

Are views presented in different viewpoints in the documentation consistent?

Are the terms that are used defined?

Are the (new) concepts defined and explained?

Is too much information included in the model?

Is the staff responsible for the documentation identified clearly?

Is it clear how the documentation will be maintained once it has been accepted?

Hämäläinen and Markkula's framework supplies a valuable basic set of questions and perspectives, but extending it to accommodate new review purposes and stakeholders is not addressed.

An *Active Design Review* is a technique for carrying out guided documentation-based reviews [Parnas 1985]. Some of the question sets provided in this report use the Active Design Review approach. Rather than invoke an all-hands room-filled design review meeting, active design reviews carefully recruit selected expert reviewers. Each reviewer is asked to review a document (or a selected subset of a document) from a particular perspective that draws upon that reviewer's special expertise. Each reviewer is given a questionnaire to guide the review. The questionnaire eschews questions that can simply be answered "yes" or "no" (e.g., "Did you find the document clear?"). Rather, the questionnaire focuses on questions that invite the reviewer to actually use the document to produce the answer (e.g., "Use the interface description for the Integrated Sensor Package module to write a short pseudo-code program that calculates the straight-line distance to the vehicle's destination.") The review team then interviews each reviewer to gauge his or her opinion of the document and how well it serves its purpose. Active design reviews tend to take more calendar days than conventional reviews, but they involve less staff time overall and produce superior results.

SEI Active Reviews for Intermediate Designs (ARID) is a method for performing a scenario-based stakeholder-centric review of a portion of architecture [Clements 2002]. The review is focused on whether the design is sufficient for the software developers who will use it. ARID is based on active design reviews and the ATAM. The elements of the ARID method could be focused on documentation to create a method to review documentation in line with the approach described in this technical note. The question set for supporting development is especially relevant. Active design reviews are a most promising starting point. For example, active design reviews call for recruiting different kinds of reviewers for different kinds of reviews. Support staff is often used, for instance, to review for document consistency and completeness and for conformance to a template. Active design reviews naturally go with the idea of a spectrum of review purposes, either as separate reviews or as multiple purposes of a single review.

Architecture-centered software project planning (ACSPP) [Paulish 2002] is another approach (like ARID) where a portion of the architecture documentation is given to the developers who are asked to use it. In this case, they are asked to take four hours to sketch an initial design of the subsystem they are tasked with developing and to fill out a sheet of metrics documenting the time and resources needed for the development effort. The question set for supporting development would be relevant for that part of the effort that involves understanding the architecture.

The Goal, Question, Metric (GQM) approach defines a measurement model based on the assumption that to measure in a purposeful way, an organization must specify goals, trace those goals to the data that are intended to define those goals operationally, and provide a framework for interpreting the data with respect to the goals [Basili 1994]. The initial steps of the approach use busi-

ness goals to identify the appropriate metrics, and the remaining steps gather measurement data and use the results for future improvements. Goal-driven measurement likewise starts with goals and questions and adds the notion of indicators to clarify the criteria of the goals' success before arriving at metrics and measurements [Goethert 2004]. The structured approach for reviewing architecture documentation follows a similar process in first defining a purpose (or goal) for the review and then formulating questions. Success indicators and some ideas about measurement are reflected in the expected answers. The goals and indicators also determine the criticality of the questions.

7 Results and Next Steps

A preliminary draft of this document was presented for comment at the *Workshop on Reviewing Architecture Descriptions* that was held at the 2008 Working IEEE/IFIP Conference on Software Architecture (WICSA 2008) (<http://www.iso-architecture.org/wicsa2008/wrad.html>). We revised the document and used it to prepare for an ATAM following the guidelines in Section 5.1. The document was also used at the University of Groningen in the Masters of Science course on software architecture and at Vrije University in the Masters course on Advanced Topics in Software Engineering. Feedback from these uses indicated that the questions helped to review the architecture documentation in a systematic way.

Next steps include exploring how the concepts illustrated in this document could be applied as a part of the Views and Beyond approach to architecture documentation [Clements 2003] or the standard ISO/IEC 42010 (originally IEEE 1471) on architecture descriptions of software-intensive systems. This technical note was written from the perspective of software and software-intensive systems, and we would also like to examine the use of this approach for more general systems.

Appendix ISO/IEC 42010

ISO/IEC 42010:2007 is a fast-track adoption by ISO of IEEE-Std 1471-2000, *Recommended Practice for Architecture Description of Software-Intensive Systems*. As part of the fast-track agreement between IEEE and ISO, the ISO adoption of the current IEEE standard is accompanied by a project to do a coordinated update to the ISO and IEEE standards. This appendix captures the material in Working Draft 4 (WD4), dated January 2009 [ISO/IEC WD4 42010:2009]. The material in this draft has undergone substantial technical review within the working group but has not been formally balloted.

The Current Standard

Figure 4 illustrates the core concepts in the current standard.

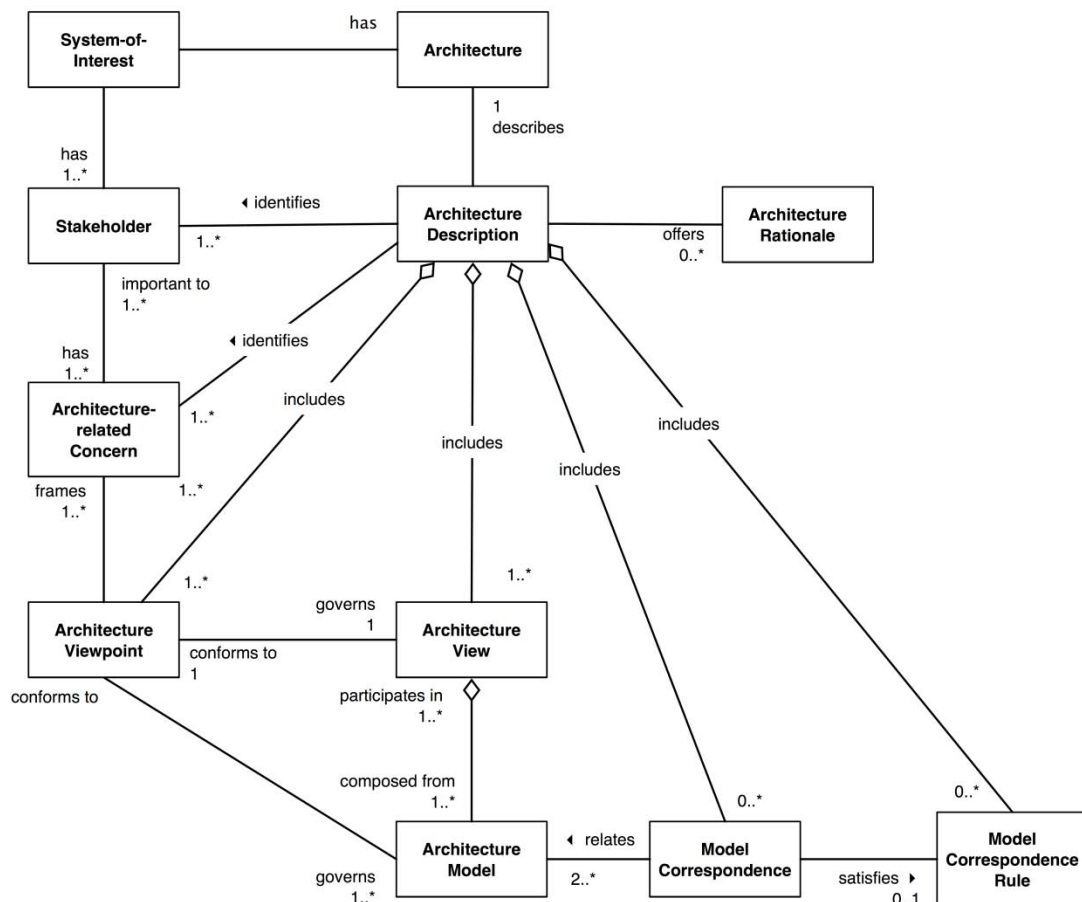


Figure 4: Core Concepts of ISO/IEC 42010:2007

An *Architecture Description* is a concrete artifact (which could be a document or repository) that documents the *Architecture* of a *System of Interest*. A *System of Interest* exists in some *Environ-*

ment (containing other systems, humans, etc.), which motivates, constrains, and interacts with the System of Interest. An Architecture Description consists of several parts as follows:

- Identification of the *Stakeholders* for the Architecture and the System of Interest
- Identification of the *Concerns* of those Stakeholders
- A set of *Architectural Viewpoints* defined so that the set of Viewpoints covers all of the Stakeholder Concerns
- A set of *Architectural Views*, exactly one for each Viewpoint
- *Architecture Rationale* to record key decisions, and so on

There are four fundamental ideas in the original IEEE 1471-2000 standard:

1. Architecture is an abstraction; architects and other stakeholders often need concrete Architecture Descriptions to deal with those abstractions.
2. Architecture Descriptions are based on multiple views. No single view is sufficient to capture an Architecture.
3. There is a need to separate Viewpoints (which capture how you want to say to describe an Architecture) from Views (which contain the descriptions of a specific Architecture). Viewpoints are reusable from system to system; views are specific to a system of interest.
4. Architecture Descriptions are motivated by Stakeholder Concerns, and one of the requirements on the notion of Viewpoints is that they exist to address these Concerns.

The Proposed Frameworks Addition

One of the goals for the ISO revision of ISO/IEC 42010:2007 was to take into consideration existing ISO standards in “architecture,” specifically GERAM [ISO 15704:2000] and RM-ODP [ISO/IEC 10746-2:1996]. The use of these standards, along with existing practice such as Kruchten’s “4+1” [Kruchten 1995], Zachman’s Architecture Framework [Zachman 1987], and even the DoD Architecture Framework [DoDAF 2007], showed there was clearly substantial practice in specifying ways to describe architectures. Often, these are referred to as *architecture frameworks*: a set of conventions for documenting architectures within a domain or stakeholder community.

In the existing IEEE 1471-2000 ontology, each of these practices could be viewed as defining a set of Viewpoints, and in fact it was the existence of approaches such as “4+1” that motivated the separation of Viewpoint from View.

Another issue not addressed by the original IEEE 1471-2000 ontology was a means to relate Views to one another in any normalized way. As an example using “4+1,” one could assert that every software element identified in the Logical view for a given architecture should be associated with at least one computing/hardware element in the Deployment view for that architecture. RM-ODP establishes a similar set of connections across the products (Viewpoints) it specifies.

Figure 5 illustrates the additions to the core IEEE 1471-2000 model to architecture frameworks.

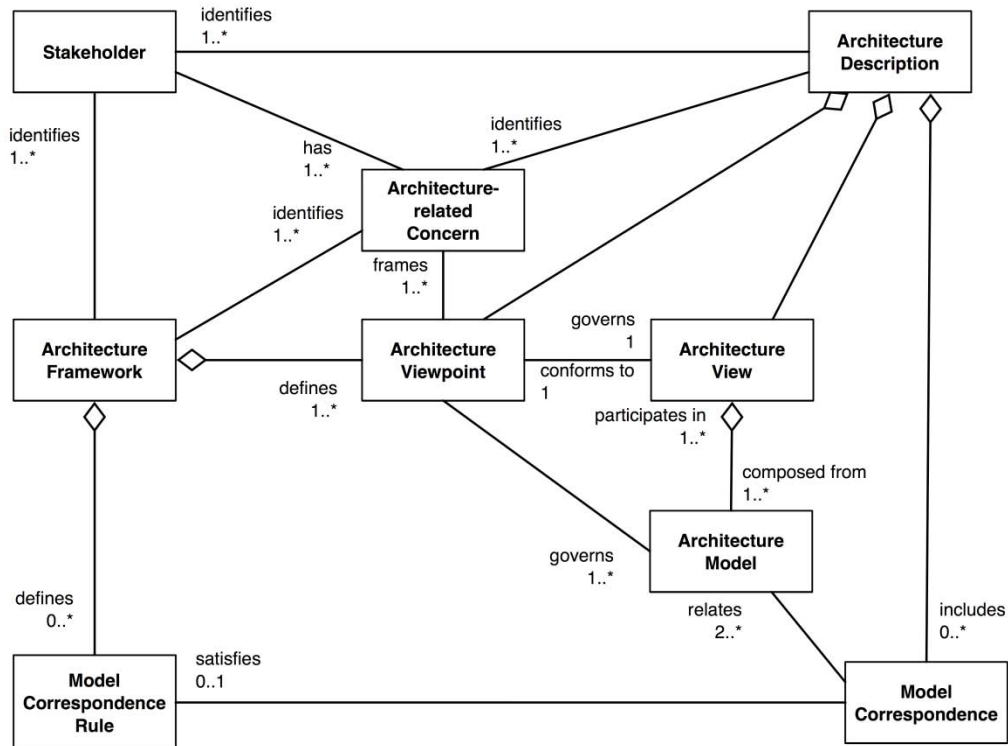


Figure 5: Additions to the Core Concepts of ISO/IEC 42010:2007

A *Model Correspondence Rule* asserts a relationship between Architecture Models within a View or across Views, such as the example from Kruchten’s “4+1 view model.” In a completed Architecture Description, each Model Correspondence Rule should be satisfied by a Model Correspondence. So a Model Correspondence for the relationship of software elements in a Logical view to computing elements in a Deployment view could be a table listing all of the software elements identified in the Logical view and for each, the computer(s) in the Deployment view where that software element will run.

ISO/IEC WD4 42010 specifies that an Architecture Framework consists of a set of Viewpoints defined by that framework’s definition and a set of Model Correspondence Rules. The Architecture Framework also identifies the architectural concerns framed by its predefined Viewpoints and the potential Stakeholders for an Architecture Description who are likely to have the Concerns identified in the Architecture Framework definition.

Applying the Standard

Although neither IEEE 1471-2000 nor the proposed ISO/IEC 42010 specifies any method for the construction of an Architecture Description (only specifying its contents), a typical approach for the application of the standard would be as follows:

1. Identify the System of Interest, its Environment, its Stakeholders, and their Concerns.
2. Select/determine the applicability of any Architecture Frameworks, which may be a design choice by the architects or a contractual requirement placed upon them.
3. Align the Architecture Framework's Stakeholders with the actual Stakeholders for the System of Interest, making sure that in particular the resulting set of Concerns is correct. Adjust the definition of Stakeholders and Concerns as appropriate.
4. Starting with those defined in the Architecture Framework, define a complete set of Viewpoints that cover the set of Concerns. Starting with those defined in the Architecture Framework, define a set of Model Correspondence Rules as necessary.
5. Implement each Viewpoint by a View in the resulting Architecture Description. Complete the Model Correspondences associated with the Model Correspondence Rules, and finally document the associated Rationale for the architecture.
6. Finally, deliver the completed Architecture Description for implementation.

As discussed in this technical note, each step in this approach can be supported by a corresponding review of the Architecture Description.

References/Bibliography

URLs are valid as of the publication date of this document.

[ANSI/EIA-748-B 2007]

American National Standards Institute and the Electronics Industry Alliance. ANSI/EIA-748-B. *Earned Value Management Systems*.
<http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI%2fEIA-748-B> (2007).

[Babar 2004]

Babar M.A. & Gorton, I. “Comparison of Scenario-Based Software Architecture Evaluation Methods,” *Proceedings of the 11th Asia-Pacific Software Engineering Conference*. NSW Australia, Nov-Dec. 2004. IEEE, 2004.

[Basili 1994]

Basili, V.; Caldiera, G.; & Rombach, H.D. “The Goals Question Metric Approach.” *Encyclopedia of Software Engineering*. John Wiley & Sons, Inc., 1994.

[Bass 2003]

Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice*, 2nd ed., (Chapter 9). Addison-Wesley, 2003 (ISBN: 978-0-321-15495-8).

[Clements 2002]

Clements, P.; Kazman, R.; & Klein, M. *Evaluating Software Architectures: Methods and Case Studies*. Addison-Wesley, 2002.
<http://www.sei.cmu.edu/library/abstracts/books/020170482X.cfm>

[Clements 2003]

Clements, P.; Bachmann, F.; Bass, L.; Garlan, D.; Ivers, J.; Little, R.; Nord, R.; & Stafford, J. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley, 2003 (ISBN: 978-0-201-70372-6). <http://www.sei.cmu.edu/library/abstracts/books/0201703726.cfm>

[Dobrica 2002]

Dobrica, L.; Niemela, E. “A survey on software architecture analysis methods.” *IEEE Transactions on Software Engineering* 28, 7 (July 2002): pp. 638-653.

[DODAF 2007]

U.S. Department of Defense Architecture Framework, Version 1.5, 2007.
[http:// www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf](http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf)

[Goethert 2004]

Goethert, W. & Sivi, J. *Applications of the Indicator Template for Measurement and Analysis* (CMU/SEI-2004-TN-024). Software Engineering Institute, Carnegie Mellon University, 2004.
<http://www.sei.cmu.edu/library/abstracts/reports/04tn024.cfm>

[Hämäläinen 2007]

Hämäläinen, N. & Markkula, J. “Quality Evaluation Question Framework for Assessing the Quality of Architecture Documentation.” *Proceedings of International BCS Conference on Software Quality Management*. University of Tampere, SQM, 2007.

[IBM 2004]

International Business Machines. *Rational Unified Process*.
<http://www-306.ibm.com/software/awdtools/rup> (2004).

[ISO/IEC 10746-2:1996]

International Organization for Standardization & International Electrotechnical Commission. *Information Technology - Open Distributed Processing - Reference Model: Foundations (ISO/IEC 10746-2)*. 1996.

[ISO/IEC 12207:2008]

International Organization for Standardization & International Electrotechnical Commission. *Systems and software engineering – Software life cycle processes (ISO/IEC 12207)*. 2008.

[ISO/IEC 15288:2008]

International Organization for Standardization & International Electrotechnical Commission. *Systems and software engineering – System life cycle processes (ISO/IEC 15288)*. 2008.

[ISO 15704:2000]

International Organization for Standardization & International Electrotechnical Commission. *Industrial automation systems - Requirements for enterprise-reference architectures and methodologies (ISO 15704)*. 2000.

[ISO/IEC 42010:2007]

International Organization for Standardization & International Electrotechnical Commission. *Systems and software engineering—Recommended practice for architectural description of software-intensive systems (ISO/IEC 42010)*. July 2007 (identical to ANSI/IEEE Std1471–2000).

[ISO/IEC CD TR 24748:2007]

International Organization for Standardization & International Electrotechnical Commission. *Systems and Software engineering - Life cycle management - Guide for life cycle management (ISO/IEC CD TR 24748)*. 2007.

[ISO/IEC WD4 42010:2009]

International Organization for Standardization & International Electrotechnical Commission. *Systems and software engineering — Architecture description (ISO/IEC WD4 42010)*. Working draft dated 26 January 2009.

Available on request from: <http://www.iso-architecture.org/ieee-1471/>

[Kruchten 1995]

Kruchten, P.B. “The 4+1 view model of architecture.” *IEEE Software* 28, 11 (November 1995):42-50.

[Parnas 1985]

Parnas, D.L. & Weiss, D.M. “Active Design Reviews: Principles and Practices,” 215-222. *Proceedings of 8th International Conference on Software Engineering*, 1985. Reprinted in Hoffman, D. and Weiss, D., *Software Fundamentals*, 2001.

[Paulish 2002]

Paulish, D.J. *Architecture-Centric Software Project Management: A Practical Guide*. Addison-Wesley, 2002.

[SARA 2002]

Final Report of the Software Architecture Review and Assessment (SARA) Group, Version 1.0, 2002. <http://philippe.kruchten.com/architecture/SARAv1.pdf>

[Schwaber 2004]

Schwaber, K. *Agile Project Management with Scrum*. Microsoft Press, 2004.

[TOGAF 1995]

The Open Group Architecture Framework, Version 8.1.1., 1995.
<http://www.opengroup.org/architecture/togaf8-doc/arch/>

[Zachman 1987]

Zachman, J.A. “A Framework for Information Systems Architecture.” *IBM Systems Journal* 26, 3 (1987).

| | | | | |
|---|---|--|--|--|
| REPORT DOCUMENTATION PAGE | | | <i>Form Approved</i> <i>OMB No. 0704-0188</i> | |
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503. | | | | |
| 1. AGENCY USE ONLY (Leave Blank) | | 2. REPORT DATE December 2009 | | 3. REPORT TYPE AND DATES COVERED Final |
| 4. TITLE AND SUBTITLE A Structured Approach for Reviewing Architecture Documentation | | | 5. FUNDING NUMBERS FA8721-05-C-0003 | |
| 6. AUTHOR(S) Robert L. Nord, Paul C. Clements, David Emery, & Rich Hilliard | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2009-TN-030 | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116 | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | |
| 11. SUPPLEMENTARY NOTES | | | | |
| 12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS | | | 12B DISTRIBUTION CODE | |
| 13. ABSTRACT (MAXIMUM 200 WORDS) This technical note proposes a structured approach for reviewing architecture documentation. Given the critical importance of architecture to software project success, it follows that the architecture cannot be effective unless it is effectively captured in documentation that allows the architecture's stakeholders to understand and use the architecture in the way it was intended. The approach does not assume a particular architecture methodology or a particular architecture documentation practice, although it was conceived in the context of the International Organization for Standardization (ISO) Recommended Practice for Architecture Description of Software-Intensive Systems and the SEI Views and Beyond approach to documenting software architectures. Like both of them, our approach is centered on the stakeholders of the artifact, engaging them in a focused, guided way to ensure that the documentation carries sufficient quality to enable them to do their jobs and to help them point out gaps and weaknesses. Our approach is not intended as a complete framework for architecture evaluation; rather it is meant to be used within such a framework, when one is available. | | | | |
| 14. SUBJECT TERMS Architecture documentation review; question set; SEI Views and Beyond; ISO/IEC 42010 | | | 15. NUMBER OF PAGES 59 | |
| 16. PRICE CODE | | | | |
| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UL | |